**Software Stability Model (SSM)**

Keynote

**Dr. Mohamed E. Fayad**

Professor of Computer Engineering

Computer Engineering Dept., College of Engineering

San José State University

One Washington Square, San José, CA 95192-0180

Ph: (408) 924-7364, Fax: (408) 924-4153

E-mail: m.fayad@sjsu.edu

http://www.engr.sjsu.edu/fayad

Dear Dr. Fayad,

I love your work and you're an amazing person.  Have you considered patenting your theory about the software development life cycle? I think that no one has ever reached to a point of generalization in a software model, but wow! This has major applications for entrepreneurship and startups to solve complex problems.

All the Best,

**Cat Volz** / Founder & Executive Producer

**The Innovation Initiative**

**Abstract**

Software Stability Model [1, 2, 3, 4] is a radically new and a disruptive innovation in the field of software engineering. It promotes a holistic approach of how any software development effort should be undertaken, different from the approaches that we see today. SSM can be used to build systems in any domain. Software Stability Model promotes a way of building software such that the software is not tied to any specific domain or application context. SSM fundamentally targets the problem of low reuse; high costs, phasing out software solutions, components with great impedance mismatch and more. Research and development done for the Software Stability Model will enable practitioners and researchers of software engineering to leverage new ways of developing software that solve these problems. The methods employed in designing software using SSM methods aid in advancing our understanding of how to do requirement analysis that is long lived and hits right at what is being aimed at. SSM will not only advance the field of software engineering but also benefit Knowledge Engineering in general. SSM concepts hold true for many disciplines, not just software. SSM fundamentally targets the problem of low reuse; high costs, phasing out of software solutions, components with great impedance mismatch and more. Research and development done for the Software Stability Model will enable the practitioners and researchers of software engineering in any discipline to look at new ways of developing software that solve these problems.

**Self-Adaptable, Self-Scalable, Stability, Self-Manageable, Easy Extensible, Self-Configurable with Unlimited Reuse:** Software that is architected and designed on the lines of SSM is done in a way that makes it and self-adaptable in nature i.e. separation of concerns between functionality sets is extremely well done. As SSM steers the designer toward a solution-strategy that employs a generic stable core, SSM enables the solution to be well-suited and adaptable to a variety of typical changes in the application context by ensuring that adaptation is limited to just the application context specific outer layer of the software longevity, high returns on investments, self-configurability, self-customizability, self-manageability, Easy Extensibility with unlimited reusability of the artifacts developed and much more.

**The advantages of SSM**

1. With SSM there is no need for re-inventing the wheels comparing to the traditional software methodology such as Spiral Model [5] and Reusable model [6] that means there is no maintenance

2. SSM creates products and components that machine and platform independent

3. SSM creates unlimited reuse of unified software architectures on Demand (USA on-Demand).

4. SSM generates assets that stable over time and have high return on investments.

5. SSM represents the core knowledge [7] of unlimited applicability that is self-adaptable, Self-Scalable, Stability, Self-Manageable, Easy Extensible, Self-Configurable with Unlimited Reuse.

6. SSM is used as a method of Domain Analysis, Analysis and Understand the software requirements that we called the problem and generates an Unlimited Solution to the problem.

**References**

[1] M.E. Fayad and A. Altman. “Introduction to Software Stability,” *Communications of the ACM,* Vol. 44, No. 9,Sept. 2001, pp. 95-98

[2] M.E. Fayad. “Accomplishing Software Stability,” *Communications of the ACM,* Vol. 45, No. 1, January2002

[3] M.E. Fayad. “How to Deal with Software Stability,” *Communications of the ACM*, Vol. 45, No. 3, March2002

[4] M.E. Fayad and S. Wu. “Merging Multiple Conventional Models in One Stable Model,” *Communications of the ACM*, Vol. 45, No. 9, 2002.

[5] Boehm, B. ["A Spiral Model of Software Development and Enhancement](http://doi.acm.org/10.1145/12944.12948)", ACM SIGSOFT Software Engineering Notes, ACM, 11(4):14-24, August 1986

[6] [Kyo C. Kang (Pohang University of Science and Technology)](http://resources.sei.cmu.edu/library/author.cfm?authorID=4443), [Sholom G. Cohen](http://resources.sei.cmu.edu/library/author.cfm?authorID=4771), [Robert R. Holibaugh](http://resources.sei.cmu.edu/library/author.cfm?authorID=4489), [James M. Perry](http://resources.sei.cmu.edu/library/author.cfm?authorID=4125), [A. Spencer Peterson](http://resources.sei.cmu.edu/library/author.cfm?authorID=4119). CMU/SEI Report Number: CMU/SEI-92-SR-004. Software Engineering Institute, January 1992.

 [7] M. E. Fayad, H. A. Sanchez, S. G .K. Hegde, A. Basia, and A. Vakil. “Software Patterns, Knowledge Maps, and Domain Analysis”. Boca Raton, FL: Auerbach Publications, Taylor & Francis Catalog #: K16540, December 2014. ISBN-13: 978-1466571433

A Pamphlet of 25-30 pages will be handed to the participants which contains this summary, 4 columns, and 2 short papers